



Europäisches Patentamt  
European Patent Office  
Office européen des brevets

⑪ Publication number:

0 171 456  
A1

⑫

## EUROPEAN PATENT APPLICATION

㉑ Application number: 84109803.1

㉑ Int. Cl. 4: G 06 F 12/14

㉒ Date of filing: 17.08.84

㉓ Date of publication of application:  
19.02.86 Bulletin 86/8

㉑ Applicant: Westheimer, Thomas O.  
P.O. Box 578  
Peterborough New Hampshire 03458(US)

㉔ Designated Contracting States:  
DE FR GB

㉑ Applicant: Hipson, Peter D.  
P.O. Box 578  
Peterborough New Hampshire 03458(US)

㉒ Inventor: Westheimer, Thomas O.  
P.O. Box 578  
Peterborough New Hampshire 03458(US)

㉒ Inventor: Hipson, Peter D.  
P.O. Box 578  
Peterborough New Hampshire 03458(US)

㉔ Representative: Meddie, Alan Leonard et al,  
FORRESTER & BOEHMERT Widenmayerstrasse 4/1  
D-8000 München 22(DE)

㉕ Computer software protection system.

㉖ In a digital computing system with a central processing unit (CPU40) and random access memory (RAM48), an improved data access limitation and protection subsystem is disclosed which protects data stored within predetermined boundaries of the RAM. An operation code detector (22) detects a unique operation code stored in the RAM and fetched by the CPU, and puts out a signal when the unique operation code is detected. An address latch (23) stores a high and a low digital boundary address put out by the CPU when the address latch is enabled by the signal from the operation code detector. An address comparator (25) compares digital addresses subsequently put out by the CPU with the stored boundary addresses and puts out a signal as the result of the comparison. The address comparator signal controls a switch which enables and disables an address transformer (43) and a bi-directional data transformer (41). A byte of data written by the CPU to the RAM is encoded by the data transformer, and a byte of data fetched by the CPU from the RAM is decoded by the data transformer; and the digital address location to which the byte of data is written and from which it is fetched is different than the digital address generated by the CPU in its normal mode of operation, if the digital address of the byte within the RAM is not greater than the high boundary address and not less than the low boundary address.

Best Available Copy

./...

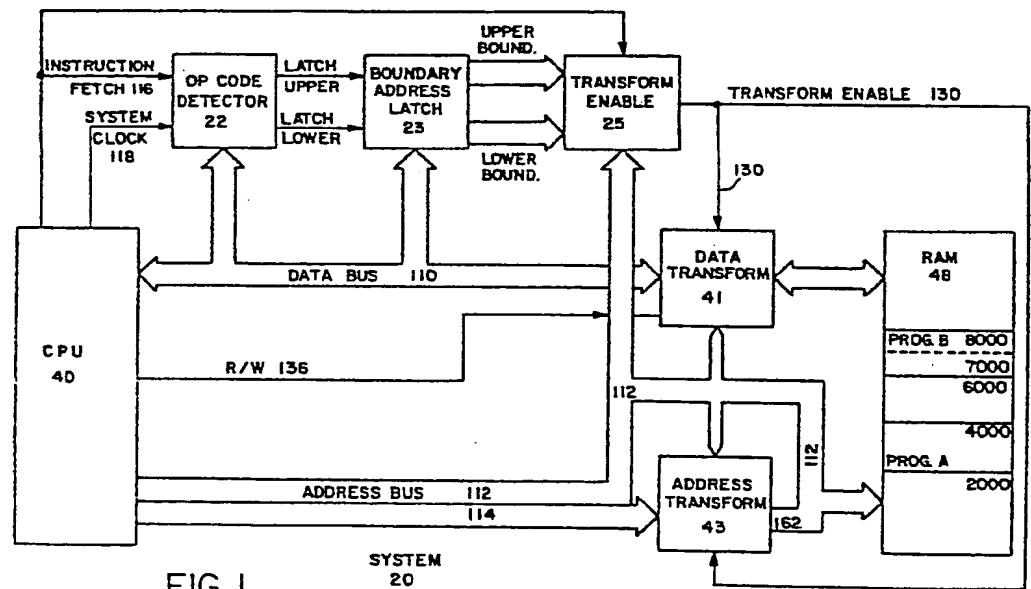


FIG. I

0171456

1 COMPUTER SOFTWARE PROTECTION SYSTEM

2

3 The present invention relates to methods and  
4 apparatus for protecting software from unauthorized access,  
5 use, and/or misappropriation. More particularly, the  
6 present invention relates to methods and apparatus for  
7 safeguarding software within a microcomputer by the use of  
8 concurrent translation of data and addresses within a  
9 predetermined protection area of main memory in accordance  
10 with an algorithm which is made unique for each  
11 microcomputer equipped with the present protection system.

12

13 The need for safeguarding software from  
14 unauthorized access, removal, and duplication has become  
15 particularly acute with the proliferation and widespread use  
16 of monolithic microprocessor-based microcomputers. The  
17 effort required by skilled programmers to design, encode,  
18 and perfect a commercially useful program can frequently be  
19 measured in terms of man-months or years. Such software is  
20 then copied onto moveable storage media, most commonly  
21 floppy diskettes, which are then distributed to authorized  
22 users. If it develops that the particular software is  
23 worthwhile, the distribution thereof becomes widespread, as  
24 does the temptation to make unauthorized copies and uses.

25

26 There have been a number of proposals for "copy-  
27 protecting" floppy diskettes. Such efforts have met with  
28 limited success, and their mere existence has led to  
29 development of software intended to break through the copy  
30 protection scheme. One significant drawback of "copy-  
31 protection" schemes is that authorized users are precluded  
32 from making back up copies of software programs and data  
33 bases.

34

35 There have been several proposals for protecting  
36 software within a computer by adapting both the hardware and  
37 the software thereof to operate only in accordance with an  
38 algorithm which is made unique for each separate computer.

0171456

1        One such proposal is found in the United States  
2 Patent No. 4,246,638, issued to William J. Thomas. The  
3 Thomas patent proposed a method for preliminarily encoding  
4 the operation code portion of an instruction as a function  
5 of the location of the instruction in main memory and as a  
6 function of the state of the computer at the time the  
7 instruction was to be executed. The drawbacks of the Thomas  
8 approach, in addition to the inherent complexity of the many  
9 uniquely connected circuit elements required to process a  
10 Thomas-encoded program, included the fact that it encoded  
11 only operation codes and, optionally, operands, but not  
12 addresses, and it dedicated three bit positions of the  
13 address bus to enable and disable the protection function.  
14 The Thomas approach thus provided only a moderate level of  
15 protection at the high price of severely restricting the  
16 size of main memory.

17

18        Another proposal is found in the United States  
19 Patent No. 4,168,396 to Robert M. Best. The Best patent  
20 describes a software protection system which deciphered  
21 preliminarily encoded information by combining the  
22 information with its address. A significant drawback of the  
23 Best approach is that it contemplated a computing  
24 environment wherein only protected software would be used  
25 with the proposed hardware. Thus, Best proposed in one  
26 preferred embodiment that the lower half of main memory be  
27 protected, thereby making it impractical, if not impossible,  
28 for the computer hardware to be usefully applied in  
29 connection with software which was not preliminarily encoded  
30 in accordance with the Best approach.

31

32        The widespread and increasing use of computers  
33 makes ever more pressing the need for a practical, cost-  
34 effective computer software protection system which does not  
35 interfere with the non-protected use of a computer and which  
36 provides for an encoding scheme of sufficient complexity to  
37 effectively eliminate the problem of unauthorized software  
38 usage. The present invention addresses this need directly

1 with a method and apparatus for temporarily protecting a  
2 limited and variable segment of computer memory. Software  
3 security is insured by transforming operation codes, data,  
4 and address locations within said segment, yet computer  
5 users have the complete and unencumbered use of their  
6 machines when the system is quiesced.

7

8         One general object of the present invention is to  
9 provide a software protection system which overcomes the  
10 limitations and drawbacks of the prior art approaches.

11

12         Another principle object of the present invention  
13 is to prevent unauthorized copying and use of software  
14 protected in accordance with the principles of the present  
15 invention.

16

17         A further object of the present invention is to  
18 provide a simplified and improved protection system which is  
19 readily implemented with available hardware elements or  
20 fabricated as a monolithic VLSI package, or as a hybrid  
21 circuit.

22

23         One more important object of the present invention  
24 is to provide an improved protection system which is totally  
25 transparent to the user and which remains quiescent until  
26 invoked by a unique operation code word, thereby enabling  
27 the full range of computer resources to be made available  
28 for use with non-protected software.

29

30         Another object of the present invention is to  
31 provide an improved protection system which enables  
32 protected and non-protected software to run concurrently on  
33 the same computer.

34

35         Yet another object of the present invention is to  
36 provide an improved protection system which enables  
37 protected software to be distributed in a universal  
38 protected format and then be automatically tailored for the

1 uniquely different protection algorithm by the user's  
2 computer in accordance with a unique tailoring program.  
3

4 Still one more object of the present invention is  
5 to provide a software protection system which protects a  
6 predetermined small yet key portion of the program, leaving  
7 major subroutines available for tailoring by each user.  
8

9 These objects are accomplished in a software  
10 protection system which includes a data transformation  
11 circuit and an address transformation circuit which are  
12 selectively enabled by a transformation enable circuit. The  
13 transformation enable circuit is initially activated by a  
14 program instruction which will normally be one of the first  
15 instructions in an encoded program. Once activated, the  
16 transformation enable circuit monitors the flow of data and  
17 addresses between the central processing unit of the  
18 computer and the computer main memory. Data and addresses  
19 which fall within a segment of memory defined by the  
20 transformation enable instruction undergo transformation,  
21 while data and addresses which fall outside said segment  
22 remain unaffected.  
23

24 The circuitry of the present invention has the  
25 advantage of relatively low cost and ease of emplacement in  
26 both new and existing computers, yet it provides a high  
27 level of protection and is effectively transparent to the  
28 user when non-encoded software is in use.  
29

30 These and other objects, advantages, and features  
31 of the present invention will be further understood and  
32 appreciated from a consideration of the following detailed  
33 description of a preferred embodiment, presented with the  
34 accompanying drawings.  
35

36 In the drawings:  
37  
38

1           Figure 1 is a simplified block drawing depicting  
2 the logical interrelationships among the various circuit  
3 elements of the present invention.

4

5           Figure 2 is a more detailed block diagram of the  
6 major electrical components comprising the present  
7 invention.

8

9           Figure 3A is a detailed block diagram of the major  
10 electrical components which comprise the operation code  
11 detector circuit of the present invention.

12

13           Figure 3B is a detailed block diagram of the major  
14 electrical components which comprise the bi-directional data  
15 transformation circuit of the present invention.

16

17           Figures 4A and 4B are timing diagrams which show  
18 the control sequences of the various circuits of the present  
19 invention.

20

21           An improved programmable digital computer system  
22 which incorporates the principles of the present  
23 invention is best understood by reference to the simplified  
24 block diagram of Figure 1. The system 20 includes a prior  
25 art central processing unit (CPU) 40 and random access  
26 memory (RAM) 48, and the circuitry of the present invention.  
27 Although the CPU 40 may comprise a type 6502 monolithic  
28 integrated circuit microprocessor made by MOS Technology,  
29 Inc., 950 Rittenhouse Road, Norristown, Pennsylvania,  
30 19401, and by other second sources, and operates with a word  
31 length of eight binary digits (bits), it will be apparent  
32 that the principles of this invention may be applied to  
33 other CPU's with other word lengths with equal success.

34

35           As can be seen in Figure 1, CPU 40 communicates  
36 with RAM 48 through data transform circuit 41 and address  
37 transform circuit 43. Both transform circuits are active  
38 only when enabled by transform enable circuit 25. When

0171456

1 disabled, data passes through data transform circuit 41 and  
2 addresses pass through address transform circuit 43 in a  
3 completely unaffected manner, so that CPU 40 is able to  
4 operate normally with non-encoded software.

5

6 Data transform circuit 41 is bi-directional, i.e.,  
7 when it is enabled, encoded data flowing from RAM 48 to CPU  
8 40 is decoded, and non-encoded data flowing from CPU 40 to  
9 RAM 48 is encoded. Data flow direction through data  
10 transform circuit 41 is controlled by the read/write (R/W)  
11 signal generated by CPU 40 and put out over the line 136.

12

13 . The lowest addressable unit of storage in RAM 48  
14 comprises eight bits, commonly referred to as a byte. Since  
15 address transform circuit 43 operates to transform only the  
16 lower eight bits of the sixteen bit address used by CPU 40,  
17 the address scrambling caused by the transformation process  
18 occurs within 256 byte intervals. The size of said interval  
19 is controlled by the number of address bits which are  
20 scrambled; thus if address transform circuit 43 were made to  
21 operate on the lower seven bits of the sixteen bit address  
22 used by CPU 40, the address scrambling would occur within  
23 128 byte intervals. Since the address space within which  
24 scrambling occurs may consist of many contiguous intervals,  
25 interval size does not present a maximum limitation to the  
26 range of addresses which may be protected. Rather, interval  
27 size is important in establishing the minimum address space  
28 which may be protected. The upper and lower addresses which  
29 define the contiguous address space within which  
30 transformation is to occur are set dynamically during  
31 program execution.

32

33 As already noted, the transform circuits 41 and 43  
34 are enabled by transform enable circuit 25. The conditions  
35 under which transform enable circuit 25 will put out a  
36 transform enable signal over line 130 are determined by  
37 operation code detector 22 and boundary address latch 23.

38

0171456

1           Operation code detector 21 is connected to data  
2 bus 110 and to instruction fetch line 110 and system clock  
3 line 118. Each time a signal is put out by CPU 40 over  
4 instruction fetch line 116, operation code detector 21  
5 examines the value on data bus 110 to see if it matches a  
6 predetermined operation code. If a match is detected,  
7 operation code detector 22 stores the next two values put  
8 out over data bus 110 into the upper and then the lower  
9 boundary address latches in boundary address latch 23.

10

11           Note that the upper and lower boundary addresses  
12 stored in boundary address latch 23 each comprise eight  
13 bits. Transform enable circuit 25 continuously samples the  
14 eight high order address bits on address bus 112 to see  
15 whether a given address put out over bus 112 falls within  
16 the upper and lower limits set in boundary address latch 23.  
17 When this condition is met, and when it is also true that  
18 the address of the last operation code sent over data bus  
19 110 fell within the upper and lower limits set in boundary  
20 address latch 23, a transform enable signal is put out over  
21 line 130. Transform enable circuit 25 "knows" when an  
22 address on bus 112 is an operation code address by also  
23 monitoring the instruction fetch signal put out by CPU 40  
24 over line 116.

25

26           When a system reset occurs, as, for example, when  
27 the system 20 is powered on, the boundary addresses in  
28 boundary address latch 25 are both set to zero. This  
29 effectively quiesces any transform operations, since no  
30 possible address on address bus 112 will cause transform  
31 enable circuit 25 to put out a transform enable signal over  
32 line 130. A user of system 20 can therefore use the system  
33 to run non-encoded software and to develop personalized  
34 software without knowledge of the operation or even the  
35 presence of the data transformation circuitry.

36

37           RAM 48 in Figure 1 depicts a non-encoded program A  
38 loaded at hexadecimal (hex) address location 2000 and

1 extending to hex address location 4000, and a protected  
2 program B loaded at hex address 6000 and extending to hex  
3 address location 8000. The protected segment of program B,  
4 represented by the shaded area in the drawing, falls between  
5 the hex address locations 6000 and 7000 within RAM 48. One  
6 of the first instructions in the program B will comprise an  
7 operation code followed by two 8 bit words which will  
8 activate operation code detector 21 and boundary address  
9 latch 23, causing the hex value 70 to be stored in the upper  
10 boundary address latch and the hex value 60 to be stored in  
11 the lower boundary address latch. Any time data is  
12 thereafter caused to be fetched from or stored into RAM 48  
13 at an address location which falls between hex 6000 and hex  
14 7000 by an operation code which is itself resident within  
15 the protected segment, transform enable circuit 25 activates  
16 transform circuits 41 and 43. Neither the portion of  
17 program B which lies outside the protected area, nor any  
18 portion of program A which may be concurrently resident in  
19 RAM 48, is affected by the transform operations.

20

21           Should an attempt be made to load and run program  
22 B on a computer system which either does not include the  
23 required data transform circuitry to decode the protected  
24 segment of program B, or which includes data transform  
25 circuitry not matched to the encoding key unique to each  
26 individual copy of program B, the computer system will  
27 encounter nonsensical instructions and addresses, thereby  
28 preventing execution of program B.

29

30           The invention thus accomplishes its primary object  
31 of providing protection for encoded software while at the  
32 same time allowing for the full and unencumbered use of the  
33 system 20 with non-encoded software. The ability to set the  
34 upper and lower boundary addresses by means of a program  
35 instruction provides protection flexibility, since the size  
36 and location of the protected program segment can be varied  
37 dynamically according to individual program needs. When a  
38 protected program terminates, it can disable any further

0171456

1 transformation by setting upper and lower boundary addresses  
2 to zero, or it can force the user to terminate the program  
3 by executing a system reset, which accomplishes the same  
4 result.

5

6 For ease of presentation, the more detailed  
7 description of the circuitry of system 20 which follows is  
8 organized by functional groups. Reference is made to Figure  
9 2.

10

11 The system 20 operates to transform data and  
12 addresses of data stored within program-defined address  
13 boundaries. The upper and lower boundary addresses are  
14 defined to and stored by the system 20 in the boundary latch  
15 circuit, comprising operation code detector 22, upper  
16 boundary latch 24, and lower boundary latch 26.

17

18 The operation code detector 22 is explained with  
19 reference to Figure 3A. Each time CPU 40 reads an operation  
20 code from RAM 48, the instruction fetch signal on line 116  
21 goes high, causing D flip-flop 52 to set the signal on line  
22 120 to the state of line 118. Line 118 is the output of 8-  
23 bit comparator 50, which compares the contents on data bus  
24 110 with a predetermined stored value, and which sets the  
25 state of line 118 high when the value on the data bus 110  
26 equals said predetermined stored value. Thus for line 120  
27 to go high, two events must occur: CPU 40 must issue an  
28 instruction fetch command, and the resulting operation code  
29 placed on data bus 110 by RAM 48 must equal the  
30 predetermined value stored in 8-bit comparator 50. When  
31 line 120 is set high, cycle counter 54 is enabled.

32

33 The operation code which causes line 120 to go  
34 high also causes CPU 40 to fetch two 8-bit words from RAM  
35 48. Coincident with each fetch, CPU 40 generates a clock  
36 pulse on line 118. The pulses on line 118, together with  
37 the enabling signal on line 120, cause the cycle counter 54  
38 first to set line 104 high and line 102 low, and then to set

0171456

1 line 102 high and line 104 low. When line 104 is set high,  
2 lower boundary latch 26 is enabled and the data byte present  
3 on data bus 110 is stored and put out on data bus 108. Bus  
4 108 will retain this value until the event sequence is  
5 repeated causing a new value to be stored in lower boundary  
6 latch 26, or until lower boundary latch 26 is cleared by a  
7 RESET signal on line 100. Similarly, upper boundary latch  
8 24 is enabled when line 102 is set high, and the data byte  
9 present on data bus 110 is put out and remains on data bus  
10 106 until a new value is caused to be stored or upper  
11 boundary latch 26 is cleared by a RESET signal on line 100.

12

13 The high-order eight bits of any address put out  
14 by CPU 40 over the 16-bit address bus comprising bus 112 and  
15 bus 114 are directed to the 8-bit comparators 28 and 30.  
16 Upper range comparator 28 has the additional input of bus  
17 106 from upper boundary latch 24, and lower range comparator  
18 30 has the additional input of bus 108 from lower boundary  
19 latch 26. Upper range comparator 28 sets output line 122  
20 high if the latched address value on bus 106 is greater than  
21 the high-order address value on bus 112. Lower range  
22 comparator 30 sets output 124 high if the latched address  
23 value on bus 108 is less than the high-order address value  
24 on bus 112. Thus when the value on bus 112 is both less  
25 than the upper boundary address stored in upper boundary  
26 latch 24 and greater than the lower boundary address stored  
27 in lower boundary latch 26, lines 122 and 124 are set high.  
28 Lines 122 and 124 feed into AND gate 32; when 122 and 124  
29 are high, line 126 is set high. Line 126 therefore is set  
30 high only when the high-order address on bus 112 falls  
31 between the upper and lower boundary addresses stored in  
32 boundary latches 24 and 26.

33

34 The output of AND gate 32 feeds into D flip-flop  
35 34 and into AND gate 36. D flip-flop 32 is clocked by the  
36 instruction fetch signal generated by CPU 40 and put out  
37 over line 116. This arrangement causes the ENCODE ENABLE  
38 output of gate 36 on line 130 to be set high when an

1 instruction fetch signal is generated by CPU 40 for an  
2 instruction stored within the address range defined by the  
3 upper and lower boundary addresses stored in the upper and  
4 lower boundary latches 24 and 26. The latching  
5 characteristic of D flip-flop 34 will also cause any  
6 subsequent address put out by CPU 40 to set the ENCODE  
7 ENABLE output of gate 36 high, regardless of the state of  
8 the instruction fetch signal on line 116, providing that the  
9 address falls within the range defined by the upper and  
10 lower boundary addresses stored in the upper and lower  
11 boundary latches 24 and 26. This condition will remain true  
12 until an operation code is fetched by CPU 40 from an address  
13 in RAM 48 which falls outside the range defined by upper and  
14 lower boundary latches 24 and 26.

15

16           The ENCODE ENABLE signal generated by gate 36  
17 controls both the data transformation circuitry and the  
18 address transformation circuitry, as will be explained  
19 later. It can be understood at this time, however, that the  
20 design of the encode enable circuitry has the following  
21 salient characteristics as depicted in Figures 4A and 4B:

22

23           - the ENCODE ENABLE signal on line 130 can only be set  
24 high by an operation code which resides within the boundary  
25 address range defined by upper and lower boundary latches 24  
26 and 26, i.e., by an operation code which is itself encoded.  
27           - once the ENCODE ENABLE signal on line 130 has been set  
28 high, any subsequent data put out or read in by CPU 40  
29 residing at an address location which falls within the  
30 boundary address range defined by upper and lower boundary  
31 latches 24 and 26 will undergo transformation by the data  
32 transformation circuitry. Equally important, data put out  
33 or read in by CPU 40 which falls outside said boundary  
34 address range will not undergo transformation. An encoded  
35 operation code can therefore reference both encoded and non-  
36 encoded data.

37

38 - once the ENCODE ENABLE signal on line 130 has been set

1 high, any subsequent address put out by CPU 40 which falls  
2 within the boundary address range defined by upper and lower  
3 boundary latches 24 and 26 will undergo transformation by  
4 the address transformation circuitry. Equally important,  
5 addresses put out by CPU 40 which falls outside said  
6 boundary address range will not undergo transformation. An  
7 encoded operation code can therefore reference both encoded  
8 and non-encoded addresses.

9

10 - once the ENCODE ENABLE signal on line 130 has been set  
11 low, no subsequent data put out or read in by CPU 40,  
12 regardless of its address, will undergo transformation by  
13 the data transformation circuitry. A non-encoded operation  
14 code can therefore reference encoded data, although this is  
15 not recommended.

16

17 - once the ENCODE ENABLE signal on line 130 has been set  
18 low, no subsequent address put out by CPU 40, regardless of  
19 its value, will undergo transformation by the address  
20 transformation circuitry. A non-encoded operation code can  
21 therefore reference an address within the address range  
22 defined by the upper and lower boundary latches 24 and 26,  
23 although this is not recommended.

24

25 The bi-directional data transformation circuit  
26 comprises ROM 42 and bi-directional gate 38. It can be seen  
27 in Figure 2 that data flows between CPU 40 and RAM 48 only  
28 through bi-directional gate 38, regardless of the data flow  
29 direction. Bi-directional gate 38 is explained with  
30 reference to Figure 3B.

31

32 Each of the individual gates depicted in bi-  
33 directional gate 38 actually represents eight identical  
34 gates, one for each bit of the 8-bit buses 110, 134, and  
35 138. Bi-directional gate 38 is controlled by the signal on  
36 line 132, which is the inverted value of the signal on line  
37 130. Line 132 is one input to each of the 8 NOR gates 56;  
38 the other input to each NOR gate 56 is one bit from the 8-

1 bit bus 136 from ROM 42. When the ENCODE ENABLE signal of  
2 line 130 is low, i.e., data is not to be transformed, the  
3 signal on line 132 is high. This will force the output of  
4 each NOR gate 56 to be low, so that all 8 bits of bus 140  
5 will be low. When the ENCODE ENABLE signal of line 130 is  
6 high, i.e., data is to be transformed, the signal on line  
7 132 is low. This will cause the output bit of each NOR gate  
8 56, which bits together comprise bus 140, to be the inverse  
9 of each corresponding input bit from bus 138.

10

11 Bus 140 feeds into the 2 sets of 8 exclusive-OR  
12 gates 60 and 68. When all 8 bits of bus 140 are low, i.e.,  
13 when ENCODE ENABLE is low, the exclusive-OR gates 60 will  
14 cause the bus 144 to contain the same value as exists on bus  
15 142. Similarly, bus 150 will contain the same value as  
16 exists on bus 148. Thus data which enters bi-directional  
17 gate 38 on bus 110 will flow in through gates 60 and out on  
18 bus 144 to bus 146 to bus 134 in a completely unaffected  
19 manner, and data which enters bi-directional gate 38 on bus  
20 134 will flow in through gates 68 and out on bus 150 to bus  
21 152 to bus 110, and will likewise remain unaffected.

22 The direction of data flow is determined by the  
23 read/write (R/W) signal generated by CPU 40 and put out over  
24 line 136. When the R/W signal is high, indicating a read  
25 operation, the signal will cause inhibit gates 64 to allow  
26 data to pass freely from bus 150 to bus 152, and thus to bus  
27 110. The same R/W signal will pass through inverter 58 to  
28 inhibit gates 62, causing these gates to block the flow of  
29 data from bus 144 to bus 146. When the R/W signal is  
30 inverted by CPU 40 to indicate a write operation, gates 64  
31 inhibit data flow from bus 150 to 152, and gates 62 allow  
32 data flow from bus 144 to 146.

33

34 When ENCODE ENABLE is high, line 132 will be low.  
35 As noted earlier, this will cause bus 140 to carry the  
36 inverted values of the corresponding bits comprising bus  
37 138. These inverted values participate in exclusive-OR  
38 operations with the corresponding bits of buses 142 or 148

1 in gates 60 or 68, respectively, depending on the direction  
2 of data flow as determined by the R/W signal over line 132  
3 from CPU 40. Thus ENCODE ENABLE will cause data flowing  
4 through bi-directional gate 38 from bus 110 to bus 134, or  
5 from bus 134 to bus 110, to undergo an exclusive-OR  
6 operation with the inverted value of bus 138.

7

8           The value of the data carried on bus 138 is  
9 determined in the following manner: ROM 42 contains eight  
10 addressable tables, each consisting of 256 addressable bytes  
11 of data. When data is to be read from or written to RAM 48  
12 by CPU 40, the address of the data is put out by CPU 40 over  
13 address buses 112 and 114. Three bits of the high-order  
14 address bus 112 are selected and passed to ROM 42 on bus  
15 156, and all eight bits of the low-order address bus 114 are  
16 passed to ROM 42 on bus 158. The three bits of bus 156 are  
17 used to select one of the eight tables stored in ROM 42, and  
18 the eight bits of bus 158 are used to select one of the 256  
19 bytes from the selected table. The value thus selected is  
20 put out on bus 138 to the bi-directional gate 38.

21

22           The address transformation circuit comprises ROM  
23 44, inverter 70, and gates 72 and 74. ROM 44 is operated in  
24 a manner identical to the operation of ROM 42 described in  
25 connection with the data transformation circuit: three bits  
26 of the high order address bus 112 and all eight bits of the  
27 low order address bus 114 are passed to ROM 44 on buses 156  
28 and 158 respectively. The three bits on bus 156 are used to  
29 select from among eight tables stored in ROM 44, and the  
30 eight bits on bus 158 are used to select from among the 256  
31 bytes of data comprising each of the eight tables. The byte  
32 so selected is put out on bus 160 to a selector circuit  
33 comprising the gates 72 and 74.

34

35           When an address is generated by CPU 40 for either  
36 a read or a write operation, the high order bits of the  
37 address pass directly to RAM 48 on address bus 112. The low  
38 order bits may also pass directly to RAM 48, or a

0171456

1 substitution value may pass to RAM 48 instead, depending on  
2 the state of gates 72 and 74. The substitution value, if  
3 chosen, is the output value of ROM 44 as hereinbefore  
4 described.

5

6 Line 130, carrying the ENCODE ENABLE signal, is  
7 fed directly to inhibit gate 74 and is fed indirectly  
8 through inverter 70 to inhibit gate 72. When ENCODE ENABLE  
9 is high, inhibit gate 72 is disabled, allowing the output of  
10 RAM 44 to pass over bus 160 to bus 162. Simultaneously,  
11 inhibit gate 74 is enabled, blocking transmission of low  
12 order address bus 114 to bus 162. The address which reaches  
13 RAM 48 will therefore consist of a non-transformed high  
14 order byte and a transformed low order byte. When ENCODE  
15 ENABLE is low, inhibit gate 72 is enabled and inhibit gate  
16 74 is disabled, thus causing the address passed to RAM 48 to  
17 consist of the non-transformed high and low order bytes of  
18 buses 112 and 114, respectively.

19

20 Having thus described an embodiment of the  
21 invention, it will now be appreciated that the objects of  
22 the invention have been fully achieved, and it will be  
23 understood by those skilled in the art that many changes in  
24 construction and circuitry and widely differing embodiments  
25 and applications of the invention will suggest themselves  
26 without departure from the spirit and scope of the  
27 invention. The disclosures and the description herein are  
28 purely illustrative and are not intended to be in any sense  
29 limiting.

30

31 The features disclosed in the foregoing  
32 description, in the following claims and/or in the  
33 accompanying drawings may, both separately and in any  
34 combination thereof, be material for realising the invention  
35 in diverse forms thereof.

36

37

38

1    CLAIMS

2

3        1. In a digital computing system including a  
4 central processing unit (CPU) capable of writing data to and  
5 reading data from a random access memory (RAM), which RAM is  
6 capable of storing and putting out data as a plurality of  
7 digitally addressable bytes under control of said CPU, and  
8 which CPU and RAM are connected by a common data bus and a  
9 common address bus, an improved data access limitation and  
10 protection subsystem for protecting data stored within  
11 predetermined boundaries of said RAM from unauthorized  
12 access comprising:

13

14            operation code detector means for detecting a  
15 unique operation code stored in said RAM and fetched by said  
16 CPU, and for putting out a signal when said unique operation  
17 code is detected;

18

19            address latch means connected to said operation  
20 code detector means for storing a high and a low digital  
21 boundary address put out by said CPU when said address latch  
22 means is enabled by said signal from said operation code  
23 detector means;

24

25            address comparator means connected to said CPU and  
26 to said address latch means for comparing digital addresses  
27 subsequently put out by said CPU with said stored boundary  
28 addresses and for putting out a signal as the result of said  
29 comparison;

30

31            address transformation means connected between  
32 said RAM and said CPU for transforming said subsequent  
33 digital addresses into different digital addresses;

34

35            bi-directional data transformation means connected  
36 between said RAM and said CPU for encoding bytes of data as  
37 said bytes are written to said RAM by said CPU, and for  
38 decoding bytes of data as said bytes are fetched from said

1 RAM by said CPU;

2

3 switch means connected to said address comparator  
4 means, to said address transformation means, and to said bi-  
5 directional data transformation means for enabling and  
6 disabling said address transformation means and said bi-  
7 directional data transformation means according to the  
8 signal put out by said address comparator means;

9

10 whereby a byte of data written by said CPU to said  
11 RAM is encoded by said data transformation means, and a byte  
12 of data fetched by said CPU from said RAM is decoded by said  
13 data transformation means, and the digital address location  
14 to which said byte of data is written and from which said  
15 byte of data is fetched is different than the digital  
16 address generated by said CPU in its normal mode of  
17 operation, if the digital address of said byte of data  
18 within said RAM is not greater than said high boundary  
19 address and not less than said low boundary address.

20

21 2. A digital computing system set forth in claim  
22 1 in which said bi-directional data transformation means  
23 comprises:

24

25 read only memory (ROM) means for storing a secret,  
26 predetermined set of data transformation bytes;

27

28 table lookup means for using selected bits of the  
29 digital address of a data byte to select a data  
30 transformation byte stored within said ROM, and for putting  
31 out said data transformation byte;

32

33 bi-directional gate means for combining said data  
34 byte with said data transformation byte in an exclusive OR  
35 operation;

36

37 whereby said data byte is transformed as a non-  
38 linear function of its digital address.

0171456

1           3. A digital computing system set forth in claim  
2 or 2 in which said address transformation means comprises:  
3

4           read only memory (ROM) means for storing a secret,  
5 predetermined set of address transformation bytes;  
6

7           table lookup means for using selected bits of the  
8 digital address of a data byte to select an address  
9 transformation byte stored within said ROM, and for putting  
10 out said address transformation byte;

11

12           bi-directional gate means for combining selected  
13 bits of said digital address with said address  
14 transformation byte in an exclusive OR operation;  
15

16           whereby said digital address is transformed as a  
17 non-linear function of its own value.

18

19           4. A method for protecting software from  
20 unauthorized use, copying, misappropriation and the like, by  
21 encoding the software with a unique code for use on a  
22 computer system having a central processing unit equipped to  
23 detect and decode said unique code, while at the same time  
24 in no way interfering with the ability of said central  
25 processor to use non-encoded software, comprising the steps  
26 of:

27

28           encoding said software with a code made unique for  
29 each said computer,

30

31           including in said software a memory boundary  
32 operation code followed by an upper and a lower memory  
33 boundary address, which boundary addresses indicate an  
34 encoded area of main memory,

35

36           loading said software into said computer for use  
37 therein,

38

0171456

1 . detecting in said computer the presence of said  
2 memory boundary operation code and thereby enabling a  
3 boundary address latch circuit,  
4

5 reading into said enabled boundary address latch  
6 circuit said upper and lower memory boundary addresses which  
7 follow said memory boundary operation code,  
8

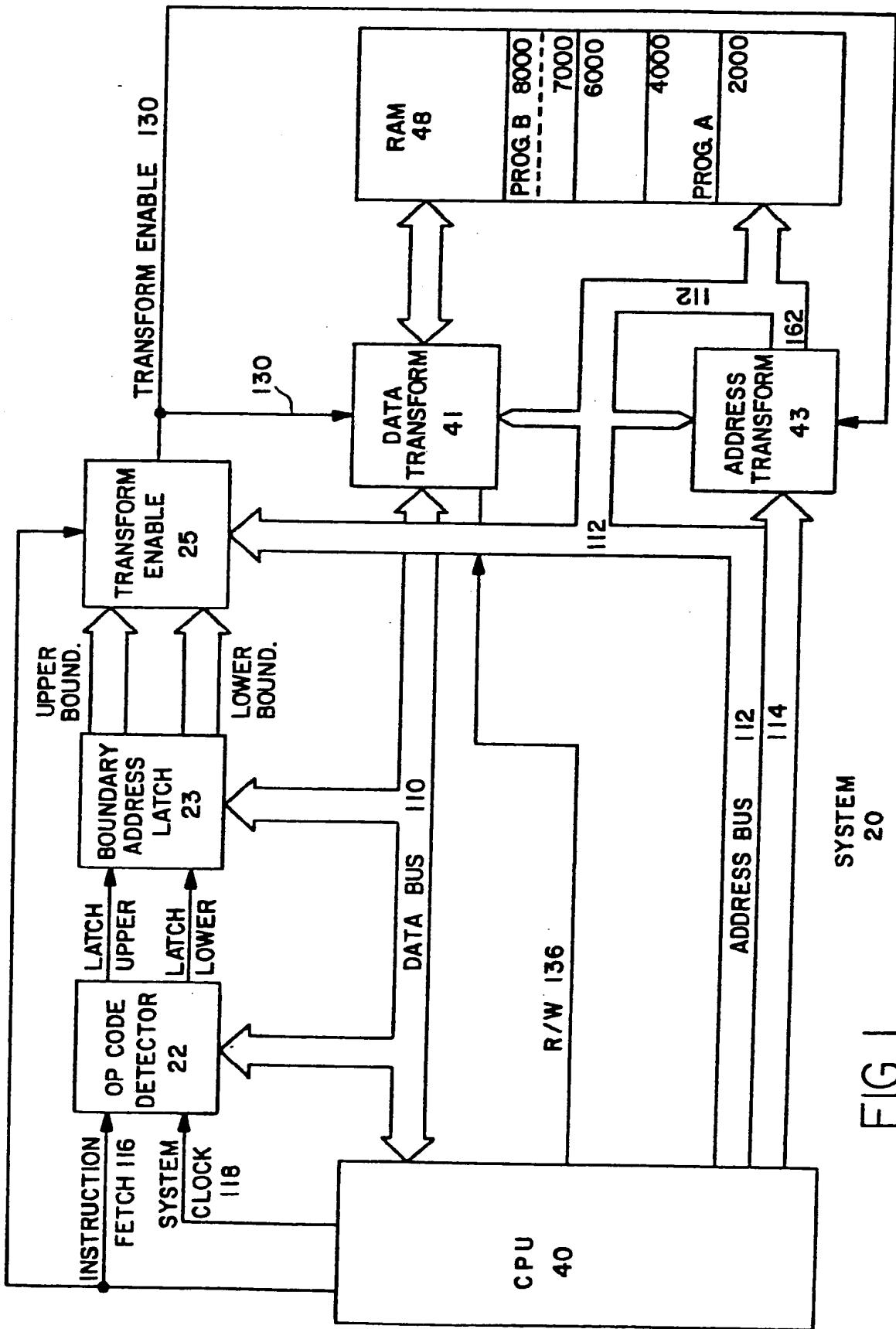
9 comparing each address of each subsequent  
10 operation code to determine whether it lies within said  
11 protected area, and if so enabling a bi-directional data  
12 transform circuit and an address transform circuit,  
13

14 decoding with said transform circuits each data  
15 word and each address word which follow said transform  
16 enabling operation code in accordance with a predetermined  
17 inverse of said unique code, providing that each such data  
18 word and each such address word also lie within said  
19 protected area,  
20

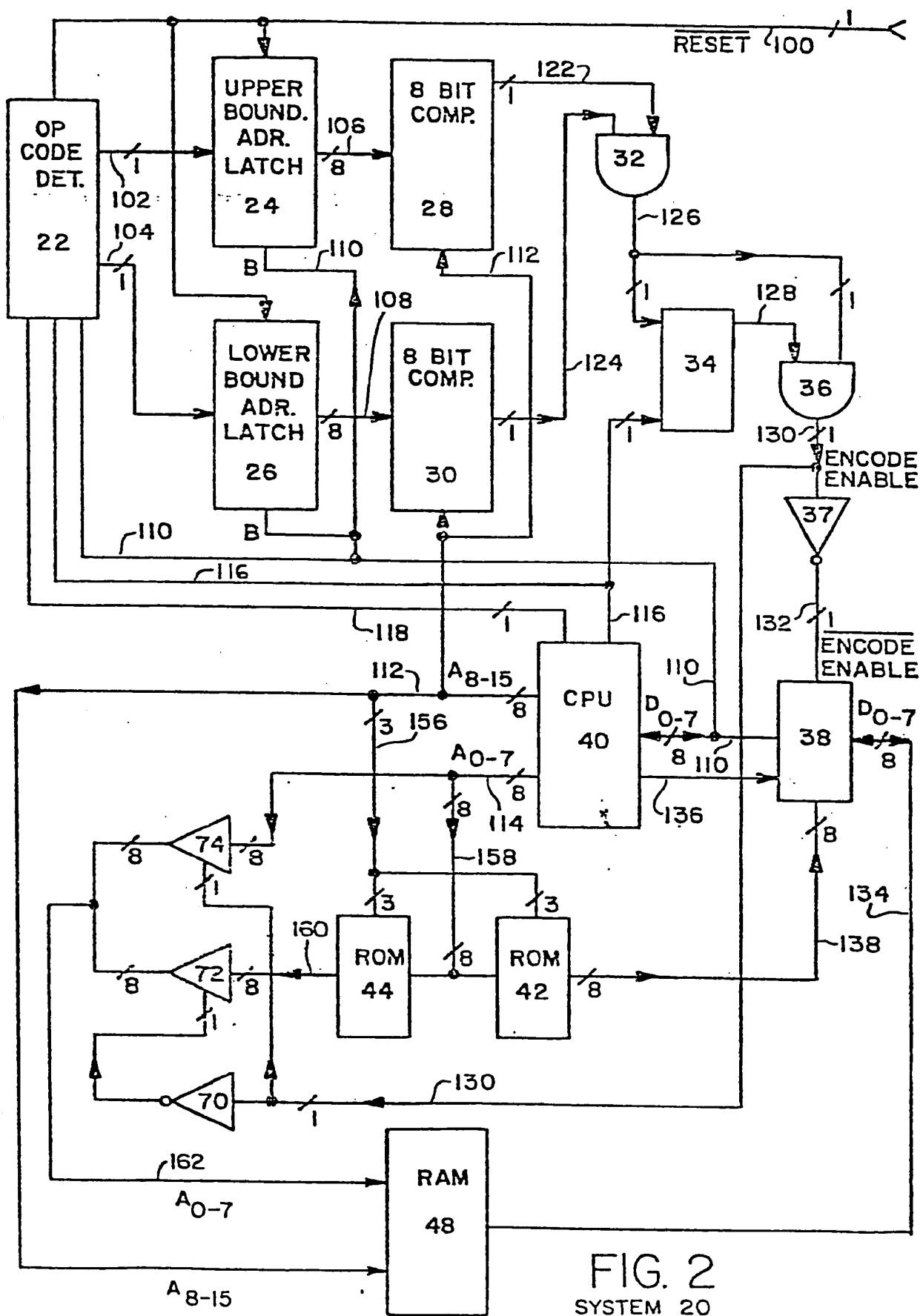
21 encoding with said bi-directional data transform  
22 circuit each data word put out which follows said transform  
23 enabling operation code in accordance with said unique code,  
24 providing that each such data word also lies within said  
25 protected area,  
26

27 disabling said transform circuits and thereby  
28 enabling said central processor unit to operate  
29 conventionally in said computer system with unencoded  
30 software.  
31

32 5. A software protection method set forth in  
33 claim 4 in which the step of disabling said transform  
34 circuits comprises including a memory boundary operation  
35 code followed by an upper and a lower memory boundary, which  
36 boundaries are equal.  
37  
38



—  
FIG.  
II

FIG. 2  
SYSTEM 20

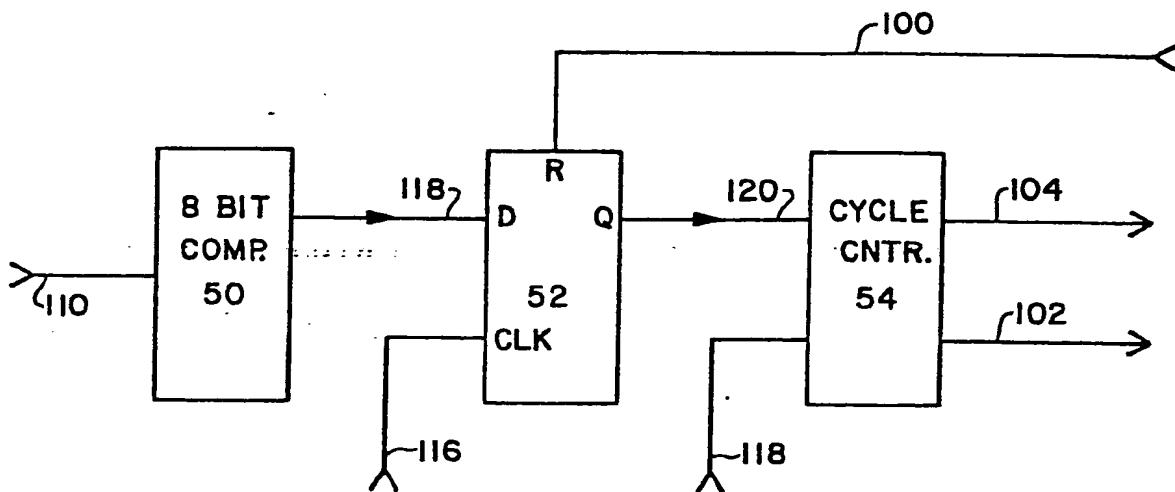


FIG. 3a

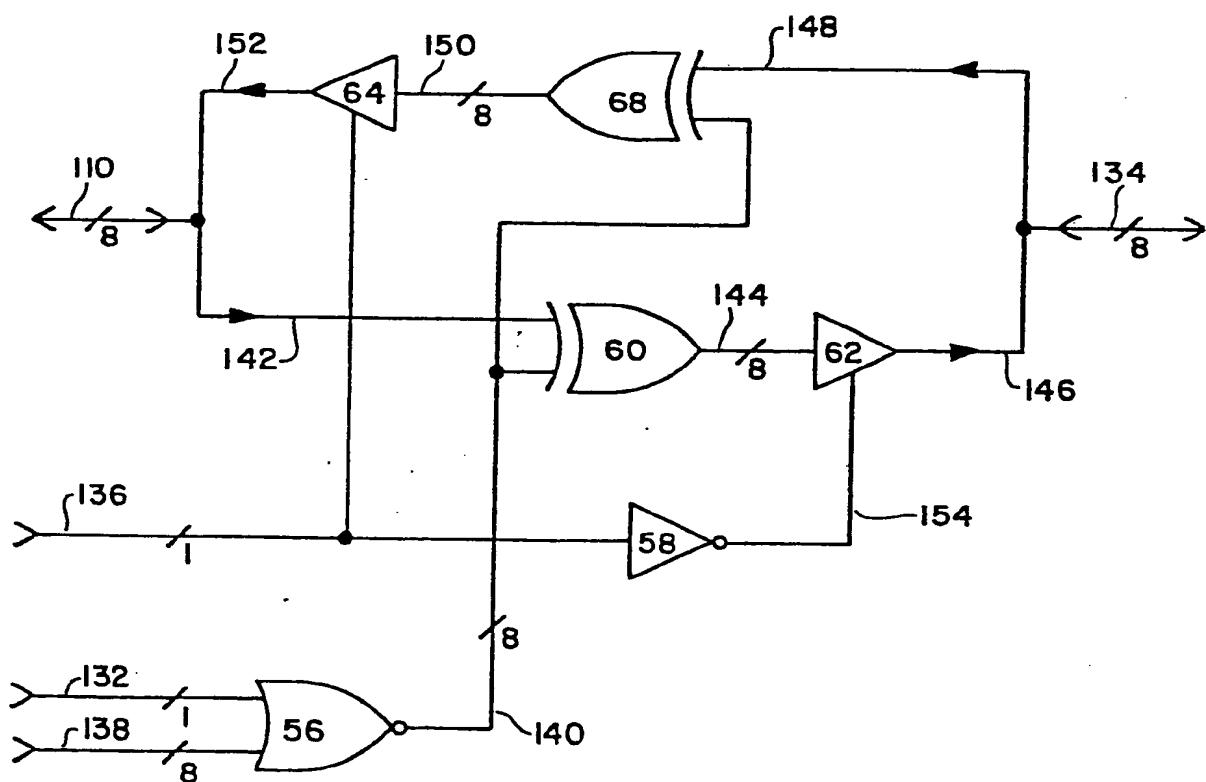


FIG. 3b

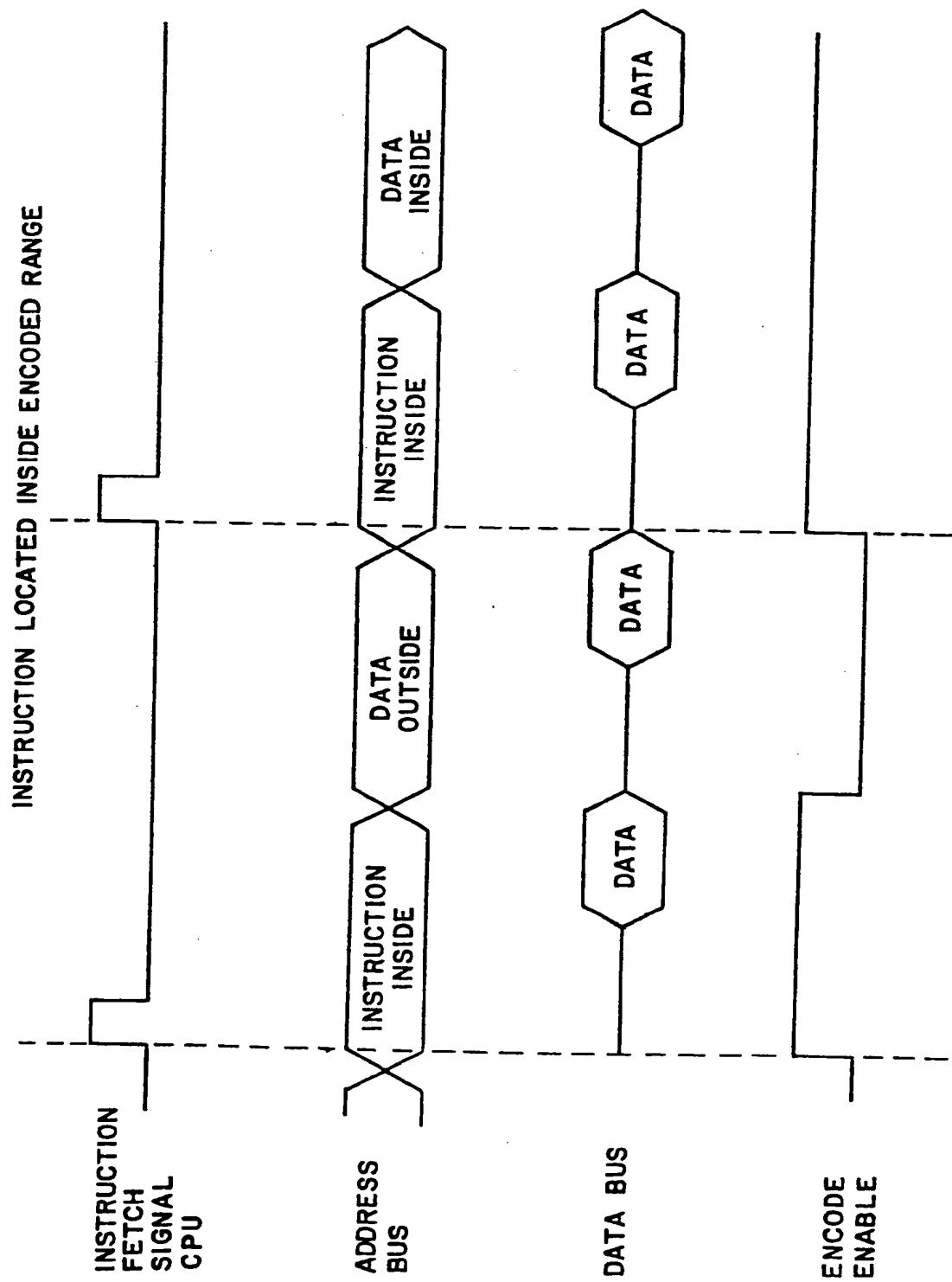


FIG. 4a

5/5

0171456

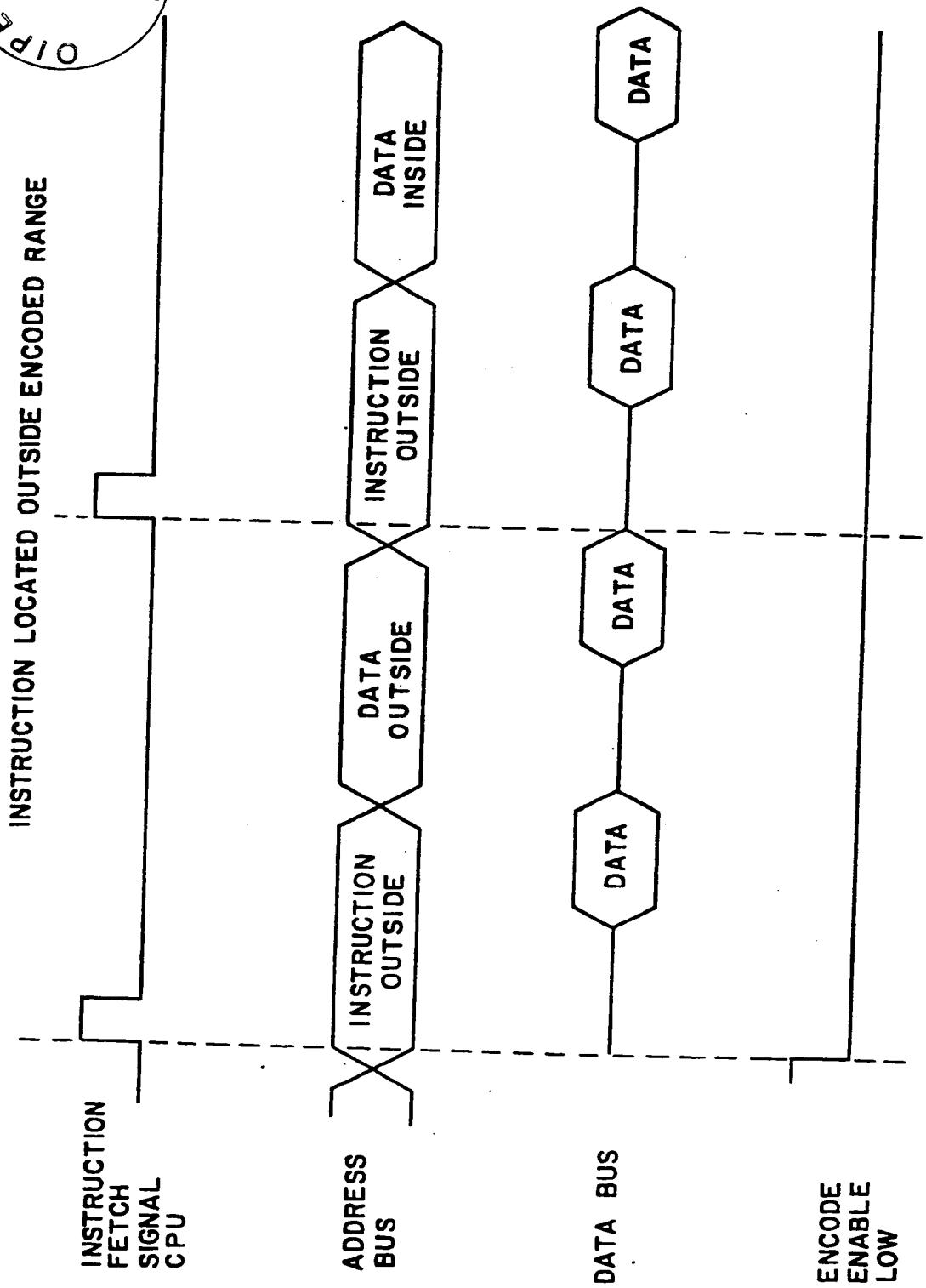
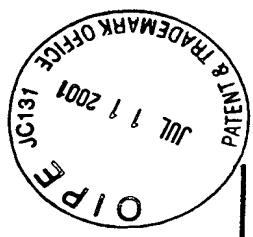


FIG. 4b

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

*THIS PAGE BLANK (68870)*